(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2001 (22.02.2001)

PCT

(10) International Publication Number
WO 01/13578 A1

(51) International Patent Classification[7]: H04L 12/18, 29/06, G06F 17/30

(21) International Application Number: PCT/US00/22413

(22) International Filing Date: 16 August 2000 (16.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/148,740 16 August 1999 (16.08.1999) US
60/184,346 23 February 2000 (23.02.2000) US

(71) Applicant: ORBLYNX, INC. [US/US]; Suite 300, 12520 Prosperity Drive, Silver Spring, MD 20904 (US).

(72) Inventors: CHEN, Hua; 6412 Coxwold Drive, Baltimore, MD 21075 (US). NGUYEN, Thuc, Dinh; 4315 Farm Oak Road, Fairfax, VA 22032 (US). ANWAR, Ibraz; 44067 Laceyville Terrace, Ashburn, VA 20147 (US). MATHUR, Anup; 18245 Rolling Meadow Way, Olney, MD 20832 (US).

(74) Agents: CARLSON, Stephen, C. et al.; McDermott, Will & Emery, 600 13th Street, N.W., Washington, DC 20005-3096 (DE).

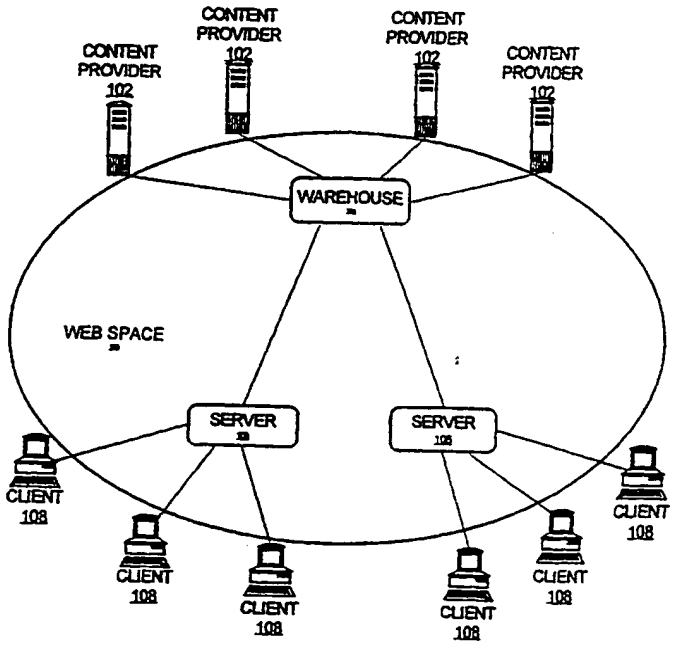
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

[Continued on next page]

(54) Title: INTERNET CACHING SYSTEM



(57) Abstract: An Internet wormhole employs a two-tier cache in the form of a warehouse system (104) that caches data from the content providers (102) on one edge of the Internet and multicasts the data to several delivery servers (106) near the users (108) on another edge of the Internet. The delivery servers (106), which may be located at the internet service providers, also cache the data for servicing a number of different users (108).

WO 01/13578 A1

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNET CACHING SYSTEM

RELATED APPLICATIONS

The present application claims the benefit of U.S. Provisional Patent Application Serial No. 60/148,760 entitled "Internet Delivery System" filed on August 16, 1999 by Hua Chen et al., the contents of which are hereby incorporated by reference in their entirety.

The present application claims the benefit U.S. Provisional Patent Application Serial No. 60/184,346 entitled "Internet Services Though a Multicast Network Overlay" filed on February 23, 2000 by H. Philip Chen et al., the contents of which are hereby incorporated by reference in their entirety.

The present application is related to the following commonly-assigned U.S. Patent Applications, the contents of all of which are hereby incorporated by reference in their entirety:

Serial No. 09/539,554 entitled "Caching Data in a Network" filed on March 31, 2000 by Hua Chen;

Serial No. _____ entitled "State-Based Caching System and Method" filed on August 16, 2000 by Hua Chen et al. (attorney docket no. 55277-023);

Serial No. _____ entitled "Community-Based Caching System and Method" filed on August 16, 2000 by Thuc Nguyen et al. (attorney docket no. 55277-024); and

Serial No. _____ entitled "Web Object Management Framework" filed on August 16, 2000 by Ibraz Anwar et al. (attorney docket no. 55277-025).

FIELD OF THE INVENTION

The present invention relates to computer networks and more particularly to an Internet wormhole architecture.

BACKGROUND OF THE INVENTION

The explosive growth in the usage of the World Wide Web is increasing demand for communications bandwidth at a global scale. For example, it has been reported that,

by the year 2001, the number of users of the World Wide Web will jump to 175 million users and the number of Web pages will increase to around 4.4 billion pages. This growth implies that the demand of bandwidth will far outpace the rate of physical network construction in the near future.

As illustrated in FIG. 5, traditional data transport on the World Wide Web employs a unicast based model from a single server system to an individual end user over the Internet. In particular, much of the Internet traffic may be characterized as the transmission of data such from a popular content provider to a large number of users at client browser systems. In FIG. 5, the World Wide Web is represented by an ever-expanding web space 500 upon whose edge various content providers 502 and client systems 504 are located. In the unicast transport model, a user at one of the client systems 504 sends a request to one of the content providers 502, which services the request and transmits a web page back to the client system 504 through the web space 500.

The unicast approach, however, will not scale to the projected size of the World Wide Web in the next few years. More specifically, as the number of users doubles, the bandwidth required to service all the users likewise doubles and the load on each of the content providers 502 doubles. Many content providers, however, cannot afford the expensive hardware and network components to adequately service the increased demand. At the same time, the number of content providers 502 is increasing, further contributing to increased load and bandwidth requirements of the World Wide Web, resulting in the "World Wide Wait." As long as the number of new users outpaces the improvements in hardware and bandwidth, such an approach is not scalable.

This lack of scalability threatens to lead to the so-called Internet meltdown as Web traffic skyrockets. As high-bandwidth audio and video streaming content, such as music and movies, becomes more popular and more frequently delivered over the Internet, the prognosis is even worse. Therefore, there is a dire need for new ways of thinking and new technologies to help alleviate the foreseeable bandwidth demands.

One conventional approach for alleviating bandwidth demands is to employ a cache at the user's browser. A cache is a local memory for temporarily storing

frequently used data. Whenever the user requests data, the cache is first inspected. If the requested data is already present in the cache, the data is served to the user directly from the cache. On the other hand, if the data is not in the cache, then the data is served from the Internet and stored in the cache.

When a cache is implemented in the user's browser, redundant requests for the same data (e.g. the second request for the same page) are served locally to the user, thereby avoiding the need to transport the content over the Internet web space 500. Browser caches, however, have a limited benefit because a single user tends to browse for new information, which is never in the browser cache. This problem is exacerbated for content providers who are constantly creating new information, because almost all of the requests they service are for the new content that will not be found in the browser caches. Thus, even with browser caches in the unicast model, content providers will ultimately have to service every user on the Internet who wants to read their content.

Accordingly, another approach is to provide a network cache for a group of users, typically associated with a particular internet service provider. With a network cache, if several users who are customers of the internet service provider wish to access the same web object across the Internet, only the request of the first user will cause the content provider 502 to supply the requested object. After the object has been supplied by the content provider 502, the object is located in the network cache for servicing subsequent users without having to go across the Internet. Unfortunately, network caches still suffer from a lack of scalability. Due to hardware constraints, a network cache can only service a fixed number of users. Thus, as the number of users double, the number of network caches will also double in the long run, causing the network traffic that populates the network cache to double. As a result, the load on the content provider systems will double as the number of users double, which is not a scalable solution as long as the increasing in user outpaces the improvements in hardware.

Another problem with browser and network caches is the stale data problem. Whenever a content provider 502 updates an existing web page, all the cached copies of the web page instantly become out of date because they do not reflect the most up-to-date version. Therefore, if a user attempts to access the latest version of the web page

but the cache has an older version, the user will get the older, stale version under the normal operation of the cache.

Various attempts have been made to address the stale data problem, but these attempts often require additional network traffic, which the provision of the cache was intended to avoid. For example, a browser could request the date of the web page at the content provider and update the cache if the web page is more recent than the cached version. In this scenario, although the amount of data is less in the message, the cache still queries the content provider 502 every time the user accesses the web page. As another example, an expiration date could be stored with the cached object, so that when the expiration date is reached, the object is assumed stale and removed from the cache. One drawback with expiration dates is that it is very difficult to determine the proper expiration date. If the expiration date is too short, then there will be unnecessary network traffic to refetch an object that has not changed after the expiration date. On the other hand, if the expiration date is too long, then the user is serviced with stale data, not with the most recent version.

Therefore, there is a need for a network architecture that can scale as the number of users on the World Wide Web skyrockets. Furthermore, there is a need for a solution to the stale data caching problem.

SUMMARY OF THE INVENTION

These and other needs are addressed by the present invention, which provides an Internet wormhole between the content providers and the users. Inspired by its namesake in the field of astrophysics (a theoretical way to beat the speed of light limitation through a short cut in the space-time continuum), an Internet wormhole is a private short cut through the Internet to quickly and efficiently transport data from the content providers to the users.

In particular, an Internet wormhole employs a two-tier cache in the form of a warehouse system that caches data from the content providers on one edge of the Internet and multicasts the data to several delivery servers near the users on another edge of the Internet. The delivery servers, which may be located at the internet service providers,

also cache the data for servicing a number of different users. Three feature cooperate and individually contribute to the Internet wormhole solution. First, the warehouse system includes a cache to reduce the load upon the content provider systems in servicing their content, thereby allowing the content providers to use less expensive hardware. Second, the multicasting from the warehouse system to the delivery servers is scalable, because even a double number of the delivery servers are still serviced in the same multicast. Finally, to address the stale data problem, the data is pushed to the delivery servers (in response to changes in the data) and overwritten in the network caches at the delivery servers, ensuring that the cached data is always fresh.

Accordingly, one aspect of the invention related to a network wormhole configuration for delivering content from a content provider system to multiple client systems. In this configuration, there are multiple delivery servers in communication with the client systems and that include a cache for storing the content. A warehouse system is in communication with the content provider system and includes a cache for storing the content and a transmission engine for multicasting the cached content by a multicast mechanism to the delivery servers and sending a command to force the respective caches to store the multicast content.

Another aspect of the invention involves a method and software for delivering content from a content provider to multiple client systems by receiving and caching the content from the content provider in a network cache; multicasting the cached content to local caches accessible to the client systems; and sending a command to force the local caches to store the multicast content.

Yet another aspect of the invention pertains to a delivery system in communication with a warehouse system and multiple client systems that includes a reliable multicast receiver configured for receiving multicast content from the warehouse system and transmitting acknowledgement messages to the warehouse system via a back channel while receiving the content. The deliver system also includes a cache configured for storing the multicast content.

Still other objects and advantages of the present invention will become readily apparent from the following detailed description, simply by way of illustration of the

best mode contemplated of carrying out the invention. As will be realized, the invention is capable of other and different embodiments, and its several details are capable of modifications in various obvious respects, all without departing from the invention. Accordingly, the drawing and description are to be regarded as illustrative in nature, and not as restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

FIG. 1 illustrates a network topology in accordance with one embodiment of the present invention.

FIG. 2 is a schematic diagram of a warehouse in accordance with one embodiment of the present invention.

FIG. 3 is a schematic diagram of an Internet delivery server according to an embodiment.

FIG. 4 depicts a computer system that can be used to implement an embodiment of the present invention.

FIG. 5 illustrates a conventional web space.

DESCRIPTION OF THE PREFERRED EMBODIMENT

A system and method for internet delivery of content are described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

NETWORK TOPOLOGY AND OVERVIEW

FIG. 1 illustrates a network topology for an Internet wormhole architecture in accordance with one embodiment of the present invention. In FIG. 1, Internet web space 100 represents the ever-increasing World Wide Web with an oval upon whose edge are located various content providers 102 and client systems 108. The content providers 102 are in communication with one end of a wormhole, namely, warehouse system 104, e.g. by a network connection that supports the Internet Protocol (IP). Although one warehouse system 104 is depicted in FIG. 1 by way of example, multiple warehouse systems 104 may be provided, and the content providers 102 may be in communication with more than one warehouse system 104.

The warehouse system 104 is also in communication with one or more delivery servers 106 at the other end of the wormhole. The delivery servers 106 are typically deployed at internet service providers (not shown), to which many client systems 108 are connected. The client systems 108 are typically personal computers, laptops, or workstations at which users execute a browser to access the Internet.

In one direction, the warehouse system 104 communicates with the delivery servers 106 by a multicast enabled infrastructure, but, in the other direction, each delivery server 106 communicates with the warehouse system 104 by a unicast back channel. The multicast enabled infrastructure can be implemented by a satellite broadcasting network, a dedicated multicast channel overlaid on the Internet, or by a multicast enabled terrestrial network such as the Internet Multicast Backbone (MBONE). In one embodiment, a reliable multicast transport mechanism is employed, in which selective acknowledgement messages are sent from the delivery server 106 via the back channel to the warehouse system 104 to ensure that every packet that was transmitted was also received. The back channel can be a typical unicast Internet connection. For example, if the delivery server 106 does not receive a packet, the warehouse system 104 is notified over the back channel of that situation (e.g. by a negative acknowledgement or time out), and the warehouse system 104 retransmits the lost packet.

As described in greater detail hereinafter, both the warehouse system 104 and the delivery server 106 employ a cache; thus, the Internet wormhole may be characterized by

a two-level cache structure. The level-one or global cache at the warehouse system 104 aggregates content from the various content providers 102 and allows that content to be served to the delivery servers 106 and ultimately to the client systems 108, directly, without burdening the content providers 102 for supplying the content. The level-two or local cache is located at the delivery server 106 and is responsible for servicing the client systems 108 with cached content without having to access the Internet web space 100.

Each of the caches can be populated by two mechanisms. The first mechanism is known as a "pull" mechanism, because, if a user requests information that is not found in the cache, the cache is responsible for fetching (or "pulling") the information from the content providers 102. In the pull mechanism, data is only cached if the data had been requested by a user. The other mechanism is known as a "push" mechanism and is described in greater detail in the commonly-assigned, co-pending application serial no. 09/539,554. Briefly, the push mechanism forces the cache to store a particular data item, regardless of whether the data item was already present in the cache. Thus, if content is pushed into the warehouse system 104 as it is updated at the content provider 102, then the cached version in the warehouse system 104 will be fresh. As a result, the content can be served up from the warehouse system 104 without needing to check with the content provider 102 to determine if the content is fresh or stale. Likewise, data that is multicast from the warehouse system 104 to the delivery server 106 is also pushed into the local cache of the delivery server 106 to ensure that the local cache contents are fresh.

These features address and meet the needs that arise from conventional network topologies. In conventional networks, there is a danger of stale data in its local caches, but with the push mechanism, the data in the warehouse system 104 and the delivery server 106 are always fresh. In conventional networks, the content providers 102 are responsible for servicing much of the data requests, thereby increasing the hardware costs for e-businesses. With the cache in the warehouse system 104, however, a content provider 102 need send only one update to the warehouse system 104, which distributes the information to its delivery servers 106 from its own cache without burdening the content provider 102. Finally, conventional networks have difficulty in scaling properly

as the number of users double, but an increased number of delivery servers 106 to receive the data on the same multicast transmission from the warehouse system 104.

CONTENT MANAGEMENT AND REGISTRATION

FIG. 2 is a schematic diagram of one implementation of a warehouse system 104. To deal with the vast proliferation of objects on the Internet, a content manager 202 and a persistent database 204 (such as a relational database) are provided to record, track, and manage all cacheable web objects handled by the system to their atomic level. The content manager 202 provides a graphic user interface front end, for example through a web browser interface, to the persistent database 204 so that an administrator may set or modify various settings and parameters to control the behavior of the system.

For example, in one embodiment, all the web content is assignable to one of several different "channels." A channel is a classification of the human-readable content of a web page, for example, news, sports, religion, humor, medicine, politics, etc. In one implementation, each web page is classified based on keywords contained therein, or signatories. For example, if a web page contains the keyword "football," then an appropriate channel for that web page would be "sports." The keywords used for classifying web pages into channels are preferably stored in persistent database 204 and manipulated by content manager 202. Thus, an administrator who wishes to set up a politics channel can specify via content manager 202 that any web page with the keyword "democrat" or "republican" is to be classified in the politics channel. A default channel may also be provided to gather unclassified objects.

Each object ever handled by the warehouse system 104, whether currently being stored in the level one cache 206 or not, is tracked by a corresponding entry in the persistent database 204 until the warehouse system 104 is able to determine that the object no longer exist (e.g. when a "404" error code is returned for an attempt to fetch the object). Each entry for an object in the persistent database 204 is characterized by its metadata, which is a collection of information about the object. In one embodiment, the metadata stored in the persistent database 204 includes the following information: a fixed-width unique identifier for the object, the next refresh time for the object, the last

cached time for the object, the last modified time, and the number of times that the object has been requested by users on the client systems 108.

The fixed-width unique identifier for the object may be calculated by hashing the URL of the object, which is a variable length string. In one embodiment, this fixed-width identifier comprises a site identifier, based on the domain name of the object, and a file identifier, based on hashing path name of the object. To disambiguate objects that happen to hash to the same value, a few bits are reserved and are uniquely assigned in case of collision. Use of the fixed-width unique identifier advantageously enables database queries on persistent database 204 to be implemented more efficiently. Thus, all URLs are internally converted to the fixed-width unique identifier, and a mapper process 208 is provided to perform the reverse calculation, from the fixed-width unique identifier to the URL. This process can be implemented by doing an indexed database lookup on a table containing the fixed-width unique identifier in one column and the variable length URL in the other column.

Another categorization of the data handled by the warehouse system 104 is based on how the object is used. At least three categories are supported: A, B, and C. The A category comprises objects that are statically determined to be popular (according to user input via content manager 202), such as the home page for yahoo.com. These and related objects are pulled into the level one cache 206. Category B relates to subscribed content, which is periodically updated and pushed into the level one cache 206 from the content providers 102. Examples of content that would classified in category B include periodically released news, articles, and stories. Category C pertains to dynamically determined popular or "hot" data that is pulled into the level one cache 206. The methodology for determining which data is hot will be described in more detail hereinafter.

In addition, a logger process 210 is provided to keep track of which content has been pushed into the level one cache 206 from the content providers 102, for example, pursuant to a category B subscription agreement. The statistics maintained by the logger process 210 are useful for billing and traffic control.

Accordingly, a persistent metadata storage system is described that is capable of managing large numbers of web objects, even those whose presence in the level one cache 206 is only transient. Furthermore, a content manager 202 is provided that allows an administrator to tailor and fine tune the content selection and classification in channels. The behavior of these features are described in more detail hereinafter.

RELIABLE MULTICASTING

As content is pushed or is otherwise entered into the level one cache 206 of the warehouse system 104, that content is ready for multicast dissemination to the delivery servers 106. In accordance with one embodiment, the multicast functionality is handled by a transmission engine 212, which includes a scheduler 214. The scheduler 214 is a process that is programmed to determine which objects and when to send to the delivery servers 106, and is typically notified when new contents is placed in the level one cache 206, for example by an active agent 228, whose functions are described hereinafter.

In one embodiment, when the scheduler 214 determines which objects to send to the delivery servers 106, the scheduler 214 prioritizes the objects based on their category, identified by accessing the persistent database 204. For example, the scheduler 214 may be configured to select a batch of objects in pre-specified proportions, such as 60% objects from Category C (hot items), 30% from Category B (subscriptions), and 10% from Category A (static popular items). The actual proportions, however, are user-configurable via the content manager 202 user interface.

Once the scheduler 214 has determined which objects belong to a bundle (typically, about 500KB to 5MB in size), the scheduler 214 spawns off a gatherer thread 216. Spawning off a gatherer thread 216 allows the scheduler 214 to concurrently operate on another bundle to send to the delivery servers 106 to permit the operating system to load balance. The gatherer thread 216 is programmed to fetch the objects identified by the scheduler 214 from the level one cache 206, and bundle the objects with their metadata extracted from persistent database 204. The bundle is then compressed and optionally encrypted and handed off to transmitter 218, which submits the bundle to a reliable multicasting mechanism 220. The reliable multicasting mechanism 220 then

transmits the objects to the delivery servers 106 via the multicast enabled infrastructure, which can be implemented by a satellite broadcasting network or by a multicast enabled terrestrial network such as the Internet Multicast Backbone (MBONE).

As used herein, the term "reliable multicasting" refers to a communication where messages are guaranteed to reach their destination complete and uncorrupted. Generally, this reliability can be built on top of an unreliable protocol by adding sequencing information and some kind of checksum or cyclic redundancy check to each message or packet. If the communication fails, the sender will be notified, for example, by a message over a back channel path from the delivery server 106 to the warehouse system 104. Various ways have been proposed for implementing reliable multicasting, but the present invention is not limited to any particular way. For example, acknowledgements could be selectively generated for messages not received in a defined timeout window, aggregated, and sent back to the warehouse system 104.

Referring to FIG. 3, which is a schematic drawing of a delivery server 106 in accordance with one embodiment of the present invention, the bundled content that has been multicast by the warehouse system 104 is received by the transmission engine 300 at each of the delivery servers 106. Specifically, the reliable multicasting mechanism 302 component of the transmission engine 300 receives the multicast packets that form the bundle and performs the appropriate error correction and acknowledge messaging to reconstitute the packets into a bundle. The bundle is then transmitted to receiver process 304, which spawns a thread for load-balancing purposes and passes the bundle to a respective scatterer processes 306.

The scatterer processes 306, upon receipt of the bundle, decrypts (if optionally encrypted) and decompresses the bundle. The content of the bundle is scanned and filtered by comparing the content with a subscription control list 308 and a block site list 310. The subscription control list 308 specifies the channels that the delivery server 106 is subscribed to. If the delivery server 106 is not subscribed to a particular channel, as specified in the persistent database 312, then all the objects received in the bundle that belong to that channel are dropped. For example, if the delivery server 106 is not subscribed to the politics channels, then all the content coming on that channel is filtered

out. The block site list 310 is an additional list that specifies which source sites are blocked from the delivery server 106. For example, if the URL of the object (or if the hashed site identifier of the object) indicates that the object comes a site that is in the block site list 310, then that object is dropped. In addition, mapper process 326, which is analogous the mapper process 208 at the warehouse system 104, may be provided to perform the conversion of the hashed URL back into the string URL, if needed.

Thus, two filtering mechanisms are provided, one that works on the channel level (subscription control list 308) and another that words on the site level (block site list 310). These lists, which are defined in the persistent database 312, can be updated by an administrator using a content manager module 324. The content manager 324 provides a user interface for modifying the adjustable and tunable parameters of the delivery server 106, including the subscription control list 308 and the block site list 310.

If the unbundled objects survive the subscription control list 308 and block site list 310 filters, then the objects are ready to be to cached at the delivery server 106. In one implementation, the scatterer process 306, which has spawned push clients 314, selects one of the push clients 314 based on the object's URL. The push client 314 forces the object to be stored in the level two cache 316, regardless of the state of the object or a previous version of the object in the level two cache 316.

Accordingly, a reliable multicasting mechanism is described in which one warehouse system 104 is capable of sending web objects simultaneously to multiple delivery servers 106. Because of this parallelism, this solution is scalable with the exploding size of the World Wide Web in contrast with conventional approaches.

COMMUNITY-BASED CACHING

One aspect of the present invention pertains to a community-based caching methodology, in which popular user content is dynamically identified when the content first starts becoming "hot" at some of the delivery servers 106. Thus, the warehouse system 104 is able to anticipate the demand for that hot content at the other delivery servers 106 and arrange for that hot content to be transferred to the other delivery servers 106 before the demand is realized.

For example, assume that a nation-wide warehouse system 104 has delivery servers 106 located in New York on the East Coast, Chicago in the Midwest, Denver in the Rocky Mountains, and San Francisco on the West Coast. In this configuration, it is likely that initial demand for content at a web site, such as the official Olympic event site, is first apparent in New York on the East Coast, because the sun rises earlier and people awake earlier. If the surge in demand holds up in Chicago, then it is likely that the content will also be wanted in Denver and San Francisco. Therefore, the warehouse system 104 arranges to have the hot content transferred to the delivery servers 106 in Denver and San Francisco, well before demand increases and when the network is less congested.

In one embodiment, statistics at the various delivery servers 106 of the users' browsing behavior is collected, processed, and transmitted to the warehouse system 104. The warehouse system 104 then analyzes the statistics to determine which content is likely to be popular, fetches the popular content, and arranges for the popular content to be multicast to the delivery servers 106. More specifically, a switch 318 is provided to intercept the user's browsing requests before the requests reach the level two cache 316. If the user's request is for non-cacheable content such as interactive telnet session, then the request is automatically and transparently rerouted to the Internet 100.

On the other hand, if the user's request is for a cacheable object, such as a web page, a graphic, or any other object such as those supported by the hypertext transfer protocol, then the request is forwarded to the level two cache 316 and logged in the persistent database 312 by a logger process 320. In particular, the logger process 320 records the number of requests for each object that a user has made. In addition, the logger process 320 also records whether the request resulted in a cache hit (i.e. the object was present in the level two cache 316) or a cache miss (i.e. the object was not present in the persistent database 312; also called Category E traffic).

The level two cache 316 upon receiving the forwarded request checks its memory to determine if the object requested by the user is present in the cache memory. If the object is present (a cache hit), then the object is served out of the cache memory of the level two cache 316 and transmitted to the client system 108. On the other hand, if the

object is not present (a cache miss), then the level two cache 316 requests the object from the Internet 100 and, upon receiving the object, caches the object within its own memory.

These statistics, e.g. the number of requests, number of cache misses, number of cache hits, are collected by the logger process 320 and stored in the persistent database 312 for a reactive agent 322 to analyze and transmit to the warehouse system 104. The reactive agent 322 is a process responsible for scouring the persistent database 312 to collect usage statistics of each object for transmission to the warehouse system 104. To save memory and bandwidth, the raw statistical data is filtered. For example, statistics of objects with a zero request count are not transmitted unless they belong to Category B (subscribed content). In the Category B case, however, even zero usage statistics are transmitted the warehouse system 104 for billing and other auditing purposes.

As another example, statistics for objects embedded in a web page such as in-line images are filtered out, because statistics for such embedded objects largely track and are thus redundant with the statistics for the web page in which they are embedded. If the web page is later determined to be popular and thus to be disseminated to the delivery servers 106, then an appropriate module at the warehouse system 104, such as a crawler process 232 described hereinafter, can ensure that the embedded objects are also disseminated along with the web page.

After filtering and on a periodic basis such as every hour, the reactive agent 322 transmits the filtered statistics to a reactive agent 222 at the warehouse system 104. The warehouse reactive agent 222 receives the statistics and performs some processing on the statistics to determine which objects are hot and what the channel should be for new objects.

A E2C converter 224 is provided to determine whether a user requested object (Category E) is hot (Category C). Various formulas may be employed to make this determination but the present invention is not limited to any particular formula. One approach is require the statistics for an object to exceed two administrator-defined thresholds. One threshold is the minimum number of delivery server 106 that has non-zero requests for the object. For example, if there are ten delivery servers 106 and this threshold is set to four, then users for at least four of the ten delivery servers 106 must

have requested that object at least one. The other threshold is a minimum total number of requests for the object among all the delivery servers 106. For example, if this second threshold is thirty, then the aggregate number of requests among all ten of the delivery servers 106 must be at least thirty. It is possible to meet both threshold with three delivery servers 106 have one request and one delivery server 106 to have twenty-seven requests for the object. These thresholds are preferably tunable by an administrator using the content manager 202 interface.

Other approaches may be employed. For example, the number of requests of an object for each delivery server 106 can be plotted in ascending order on a histogram. From area of the histogram, the center of mass is calculated and objects having histograms with more central masses are given priority over those who center of mass is skewed.

Another module is the channelizer 226, which detects whether an object is new to the warehouse system 104. If the object is new, then the channelizer 226 inspects the object to determine which channel the new object should belong to. For example, a web page can be parsed for keywords that have been preset by the administrator at the content manager 202 interface. If the keywords are found in the new object, then the new object is classified to the channel that the keywords correspond to. In the previous example, if the keyword "democratic" or "republican" is found in the object, then the object is classifies as belonging to the politics channel.

After the object has been ascertained to be "hot" by the E2C converter 224 and channelized by channelizer 226, the object is cached in the level one cache 206 for delivery to the delivery servers 106 in a subsequent multicast transmission.

Accordingly, a mechanism for community-based caching is disclosed in which information for the delivery servers 106 are gathered and processed to dynamically identify new content that is likely to be popular with the users downstream of the warehouse system 104.

STATE-BASED CACHING

Another aspect of the invention exploits the persistent metadata storage to perform state-based caching of objects at the warehouse system 104, even for objects that are not currently stored in the level one cache 206. Accordingly, an active agent 228 is provided that is configured to perform two kinds of state-based caching based on objects whose metadata is stored in the persistent database 204.

One kind of state-based caching is supplied by the refresher process 230, which periodically examines the persistent database 204 to determine if any objects (i.e. those that have been pull-cached in Categories A and C) need to be refreshed. In one embodiment, a next refresh time is calculated for each object as follows. If the object has an expiration time, then the next refresh time is set to expiration time. On the other hand, if the object does not have an expiration, then the next refresh time (NRT) is based on a relationship between the last cached time (LC), the last modified time (LM), and a tunable parameter (k). In one implementation the next refresh time (NRT) is calculated to equal the sum of the lack cached time and k times the difference of the last cached time and the last modified. The value of k, which ranges from 0 to 1, is preferably empirically determined for the particular implementation environment and is readily tuned through the content manager 202 interface. During the operation, the refresher process 230 sorts the objects by the next refresh time and select those objects with the nearest next refresh times. These objects are refreshed, causing an updated version to the supplied to the level one cache 206 and are scheduled for subsequent multicast distribution.

Another state-based caching technique involves object discovery by a crawler process 232. Stemming from the realization that there is locality in web access because users tend to follow links from one page to another, the crawler process 232 is provided to periodically retrieve registered content URLs from the persistent database 204 that need to be crawled and that have not been marked as being crawled. Typically this content includes Category A (statically popular) traffic and newly-added Category C (dynamically popular) traffic. Then the crawler process 232 fetches the objects at the URLs and parses the HTML code of the objects to find links. Preferably, a level

parameter is employed to specify the number of links deep for each page the crawler process 232 is permitted to crawl. In another feature, URLs that refer to other sites than the source page are not processed.

As new URLs are discovered by the crawler process 232, they are checked to determine if the objects at the URLs are cacheable and if so, whether the objects meet the cost-to-value criterion and are within the same domain. Various cost-to-value criteria may be employed in various embodiments of the present invention. One criterion is as follows: points are awarded to the object if the object exceeds a pre-specified size (e.g. 7KB), exceeds a pre-specified latency (e.g. 8s), or belongs to a pre-specified type (e.g. mime/html). If the number of points exceed another pre-specified value, then the object is deemed to have passed the cost-to-value criterion and is added to the level one cache 206 and the list of crawlable objects. To avoid infinite loops, objected that have been crawled are marked in the persistent database 204 to indicate that fact. If an object, however, was refreshed by the refresher process 230, then that object may be recrawled.

Accordingly, a state-based caching technique is disclosed in which objects can be refreshed or crawled even if they have been flushed from the cache.

HARDWARE OVERVIEW

FIG. 4 is a block diagram that illustrates a computer system 400 upon which an embodiment of the invention may be implemented. Computer system 400 includes a bus 402 or other communication mechanism for communicating information, and a processor 404 coupled with bus 402 for processing information. Computer system 400 also includes a main memory 406, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 402 for storing information and instructions to be executed by processor 404. Main memory 406 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 404. Computer system 400 further includes a read only memory (ROM) 408 or other static storage device coupled to bus 402 for storing static information and instructions for processor 404. A storage device 410, such as a

magnetic disk or optical disk, is provided and coupled to bus 402 for storing information and instructions.

Computer system 400 may be coupled via bus 402 to a display 412, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 414, including alphanumeric and other keys, is coupled to bus 402 for communicating information and command selections to processor 404. Another type of user input device is cursor control 416, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 404 and for controlling cursor movement on display 412. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

The invention is related to the use of computer system 400 for internet content and delivery services. According to one embodiment of the invention, internet content and delivery services is provided by computer system 400 in response to processor 404 executing one or more sequences of one or more instructions contained in main memory 406. Such instructions may be read into main memory 406 from another computer-readable medium, such as storage device 410. Execution of the sequences of instructions contained in main memory 406 causes processor 404 to perform the process steps described herein. One or more processors in a multi-processing arrangement may also be employed to execute the sequences of instructions contained in main memory 406. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 404 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media include, for example, optical or magnetic disks, such as storage device 410. Volatile media include dynamic memory, such as main memory 406. Transmission media include coaxial cables, copper wire and fiber optics, including the wires that comprise bus 402. Transmission media can also take the

form of acoustic or light waves, such as those generated during radio frequency (RF) and infrared (IR) data communications. Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, any other magnetic medium, a CD-ROM, DVD, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 404 for execution. For example, the instructions may initially be borne on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 400 can receive the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector coupled to bus 402 can receive the data carried in the infrared signal and place the data on bus 402. Bus 402 carries the data to main memory 406, from which processor 404 retrieves and executes the instructions. The instructions received by main memory 406 may optionally be stored on storage device 410 either before or after execution by processor 404.

Computer system 400 also includes a communication interface 418 coupled to bus 402. Communication interface 418 provides a two-way data communication coupling to a network link 420 that is connected to a local network 422. For example, communication interface 418 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 418 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 418 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

Network link 420 typically provides data communication through one or more networks to other data devices. For example, network link 420 may provide a

connection through local network 422 to a host computer 424 or to data equipment operated by an Internet Service Provider (ISP) 426. ISP 426 in turn provides data communication services through the worldwide packet data communication network, now commonly referred to as the "Internet" 428. Local network 422 and Internet 428 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 420 and through communication interface 418, which carry the digital data to and from computer system 400, are exemplary forms of carrier waves transporting the information.

Computer system 400 can send messages and receive data, including program code, through the network(s), network link 420, and communication interface 418. In the Internet example, a server 430 might transmit a requested code for an application program through Internet 428, ISP 426, local network 422 and communication interface 418. In accordance with the invention, one such downloaded application provides for internet content and delivery services as described herein. The received code may be executed by processor 404 as it is received, and/or stored in storage device 410, or other non-volatile storage for later execution. In this manner, computer system 400 may obtain application code in the form of a carrier wave.

While this invention has been described in connection with what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

WHAT IS CLAIMED IS:

1. A network wormhole configuration for delivering content from a content provider system to a plurality of client systems, comprising:

a plurality of delivery servers including respective caches configured for storing the content, each of the delivery servers in communication with one or more of the client systems; and

a warehouse system, in communication with the content provider system, including a cache configured for storing the content and a transmission engine configured for multicasting the cached content by a multicast mechanism to the delivery servers and sending a command to force the respective caches to store the multicast content.

2. The network wormhole configuration according to claim 1, wherein the cache of the warehouse system is adapted to receive a command from the content provider to force the cache of the warehouse system to store the content.

3. The network wormhole configuration according to claim 1, further comprising a plurality of back channels respectively between the delivery servers and warehouse system for transmitting an acknowledgement message from the delivery servers via a corresponding one of the back channels to the warehouse during a course of the multicasting, wherein the multicasting is reliable.

4. The network wormhole configuration according to claim 3, wherein the multicast mechanism includes a satellite delivery system.

5. The network wormhole configuration according to claim 3, wherein the multicast mechanism includes a dedicated multicast channel overlaid on the Internet

6. The network wormhole configuration according to claim 1, wherein the transmission engine is configured to execute a scheduler process for bundling a plurality of web objects for multicast transmission to the delivery servers.

7. A method for delivering content from a content provider to a plurality of client systems, comprising the steps of:

receiving and caching the content from the content provider in a network cache;

multicasting the cached content to a plurality of local caches accessible to the client systems; and

sending a command to force the local caches to store the multicast content.

8. The method according to claim 7, further comprising the steps of:

receiving a command from the content provider to push the content into the network cache; and

in response to the command, forcing the network cache to store the pushed content.

9. The method according to claim 7, further comprising the step of transmitting acknowledgement messages via a back channel during a course of the multicasting.

10. The method according to claim 9, wherein the step of multicasting includes the step of broadcasting over a satellite.

11. The method according to claim 7, further comprising the step of bundling a plurality of web objects for the multicasting.

12. A warehouse system in communication with a content provider system and a plurality of delivery servers, comprising:

a cache configured for storing the content; and

a transmission engine configured for multicasting the cached content by a multicast mechanism to the delivery servers and forcing the respective caches to store the multicast content.

13. A computer-readable medium bearing instructions for facilitating the delivery of content from a content provider system to a plurality of client systems, said instruction being arranged upon execution thereof to cause one or more processors to perform the steps of:

receiving and caching the content from the content provider in a global cache;

multicasting the cached content to a plurality of local caches accessible to the client systems; and

sending a command to force the local caches to store the multicast content.

14. A delivery system in communication with a warehouse system and a plurality of client systems, comprising:

a reliable multicast receiver configured for receiving multicast content from the warehouse system and transmitting acknowledgement messages to the warehouse system via a back channel while said receiving; and

a cache configured for storing the multicast content.

15. The delivery system according to claim 14, wherein the cache is further configured to force storage of the multicast content in response a push command.

16. A computer-readable medium bearing instructions for facilitating the delivery of content from a content provider system to a plurality of client systems, said instructions being arranged upon execution thereof to cause one or more processors to perform the steps of:

receiving multicast content from a warehouse system in communication with the content provider system;

transmitting acknowledgement messages to the warehouse system via a back channel during said receiving; and

storing the multicast content.

17. The computer-readable medium according to claim 16, said instructions being further arranged upon execution thereof to cause the one or more processors to perform the step of forcing storage of the multicast content in response a push command.

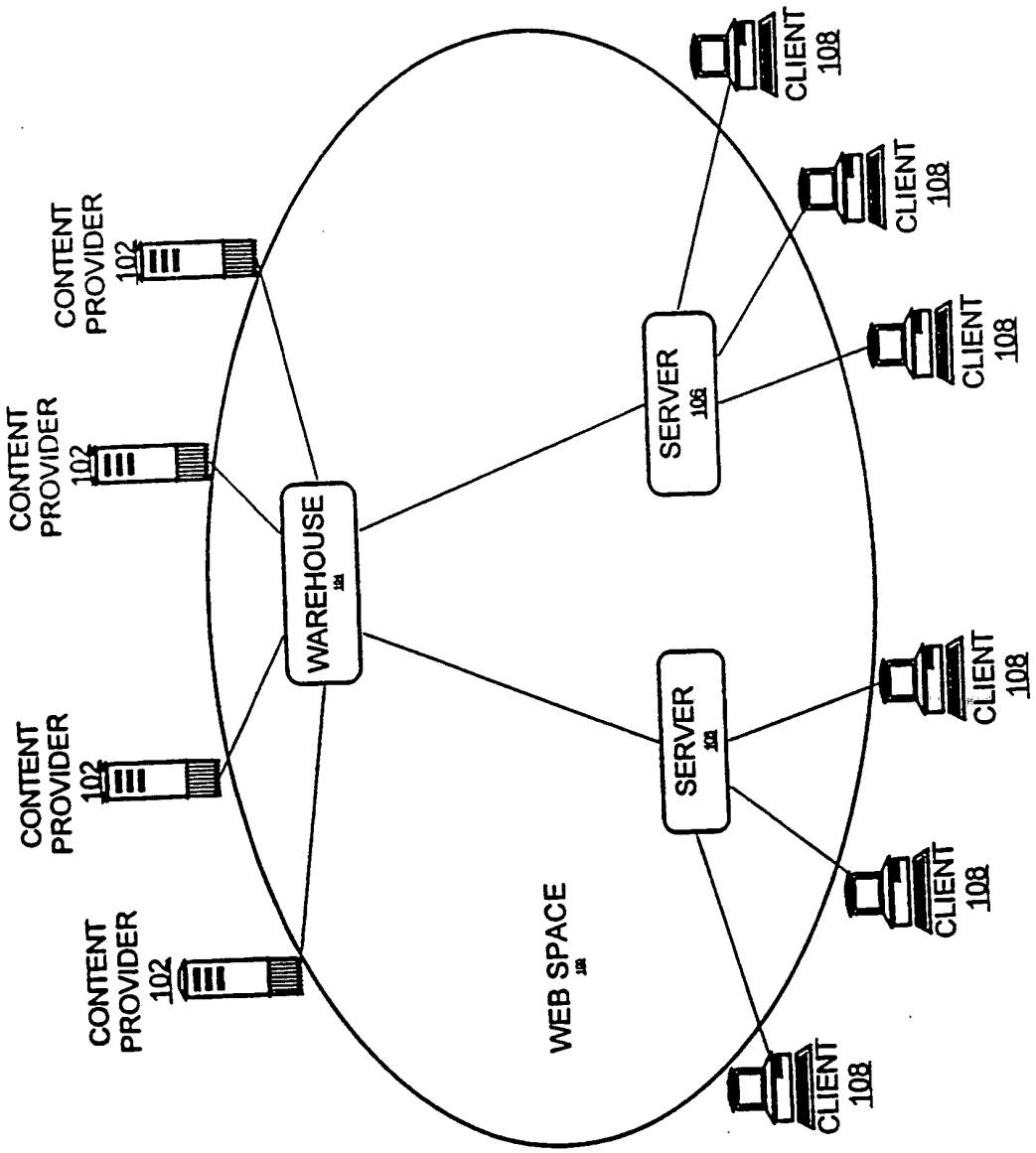


FIG. 1

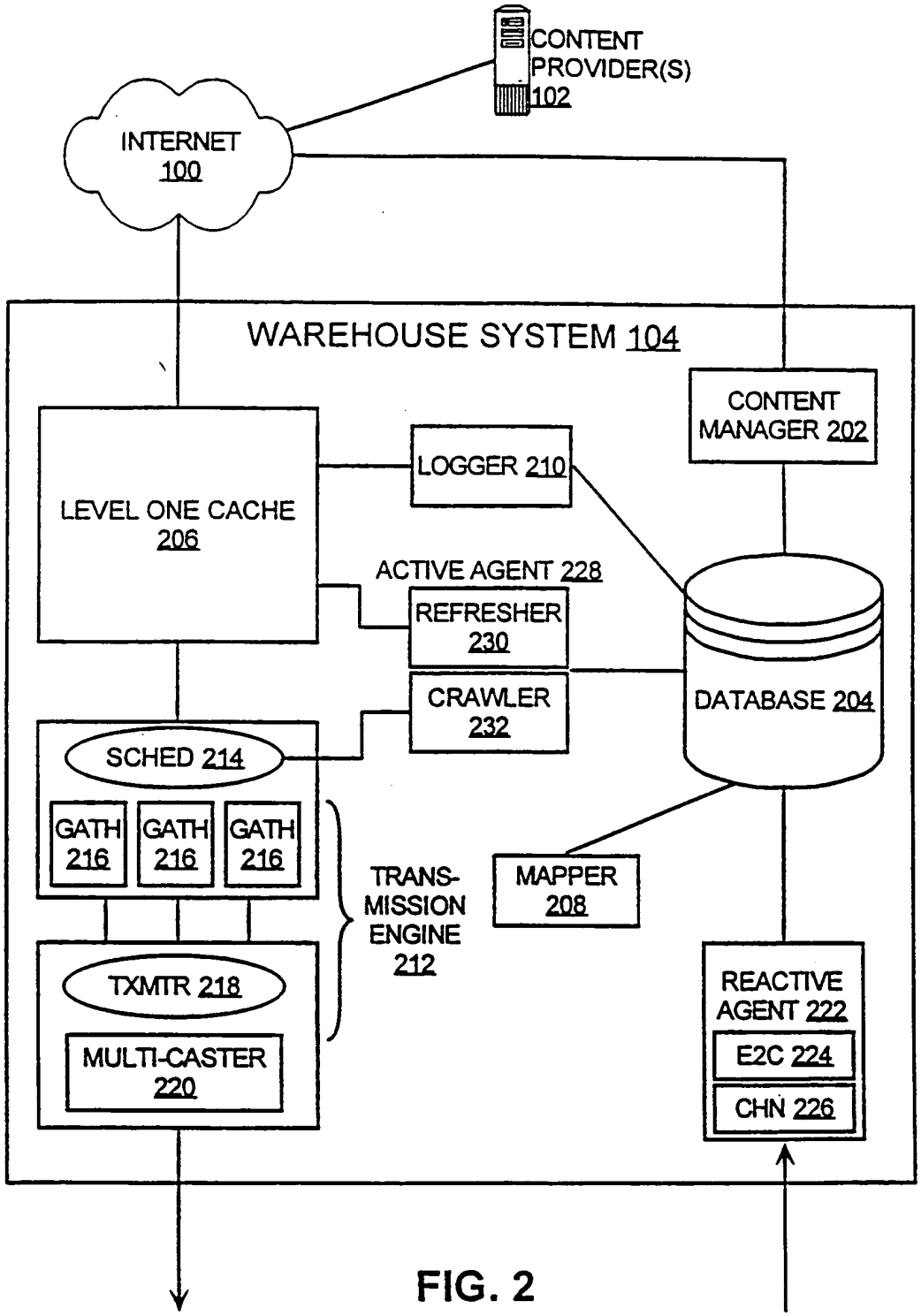


FIG. 2

WO 01/13578 PCT/US00/22413

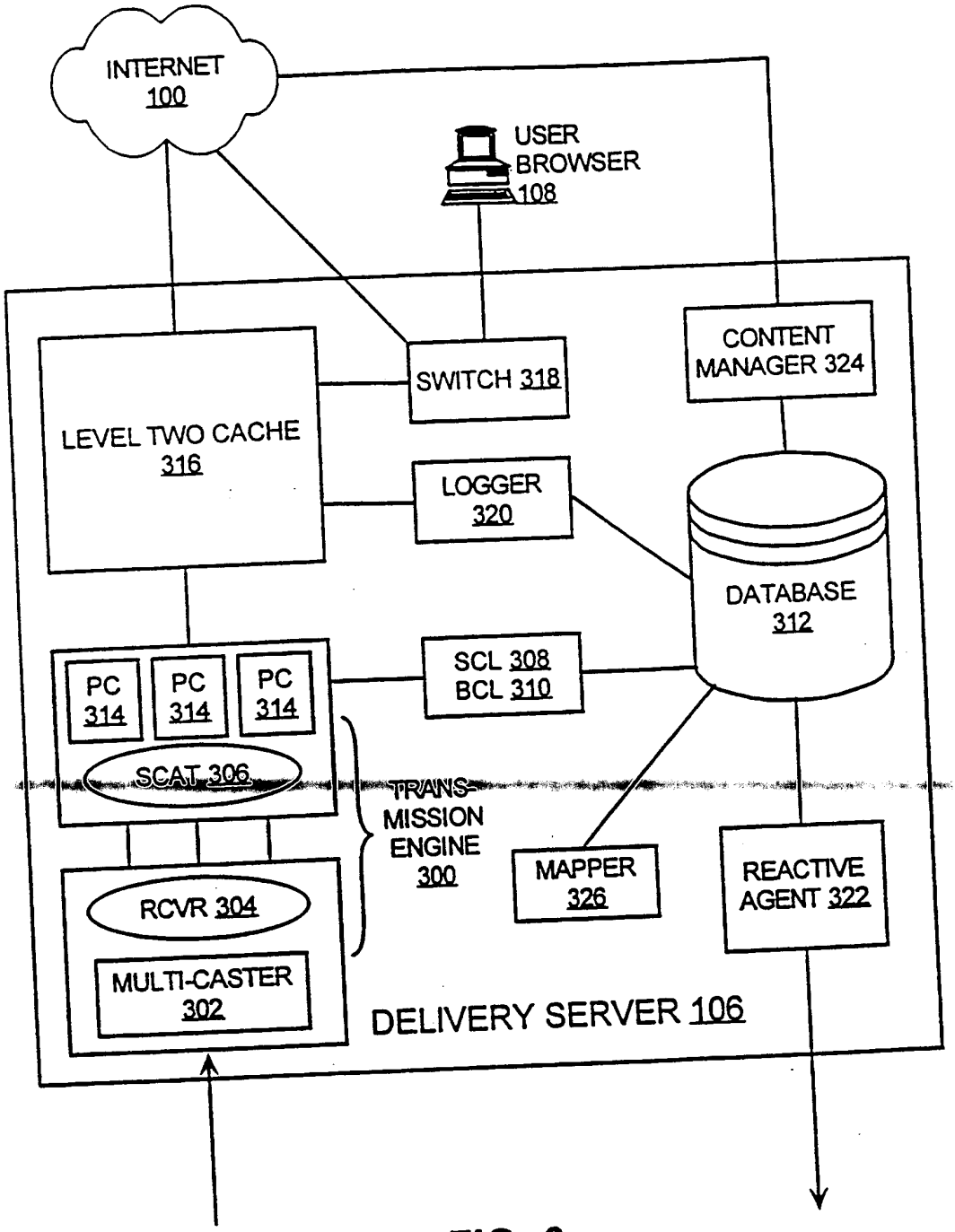


FIG. 3

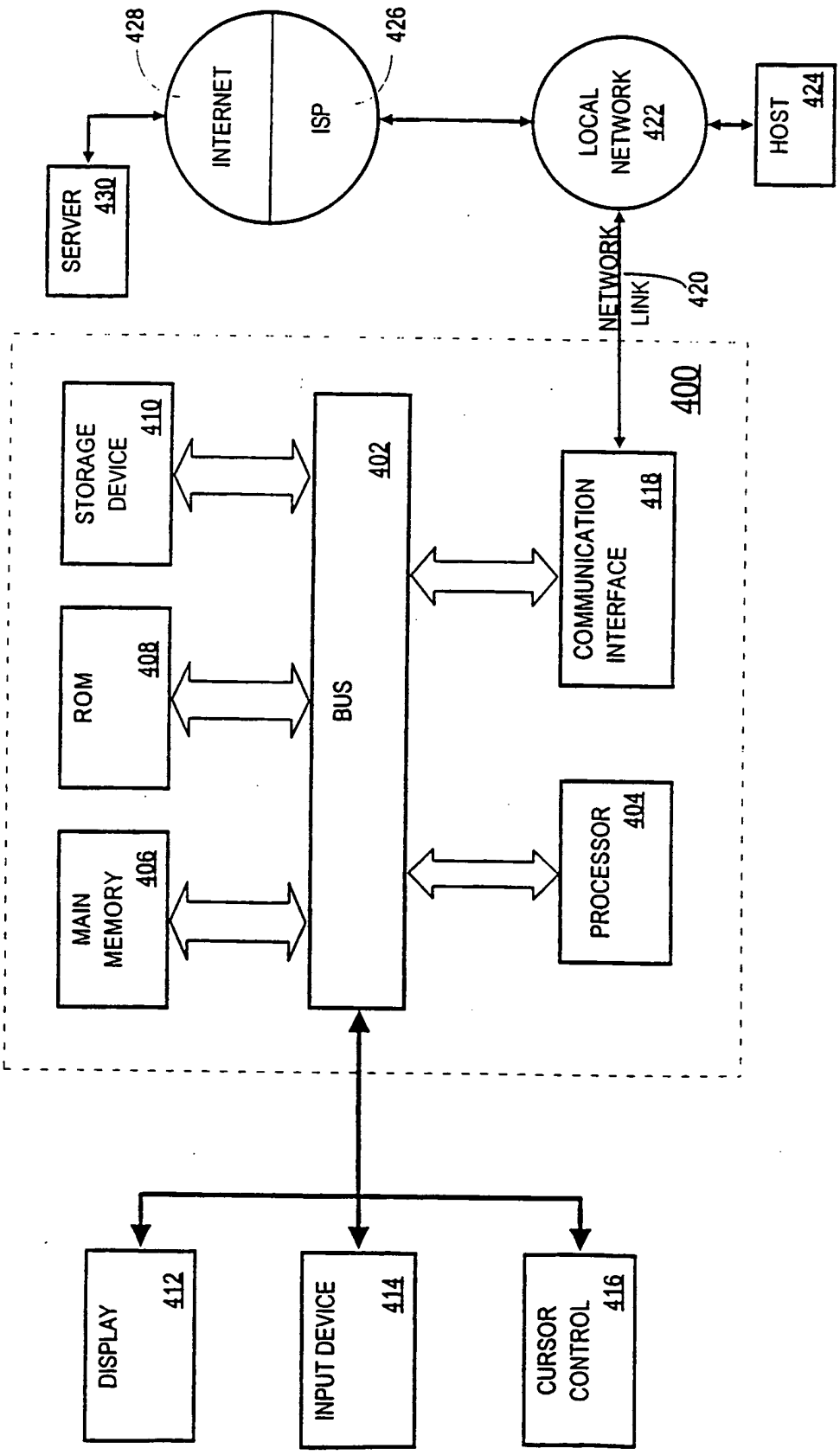


FIG. 4

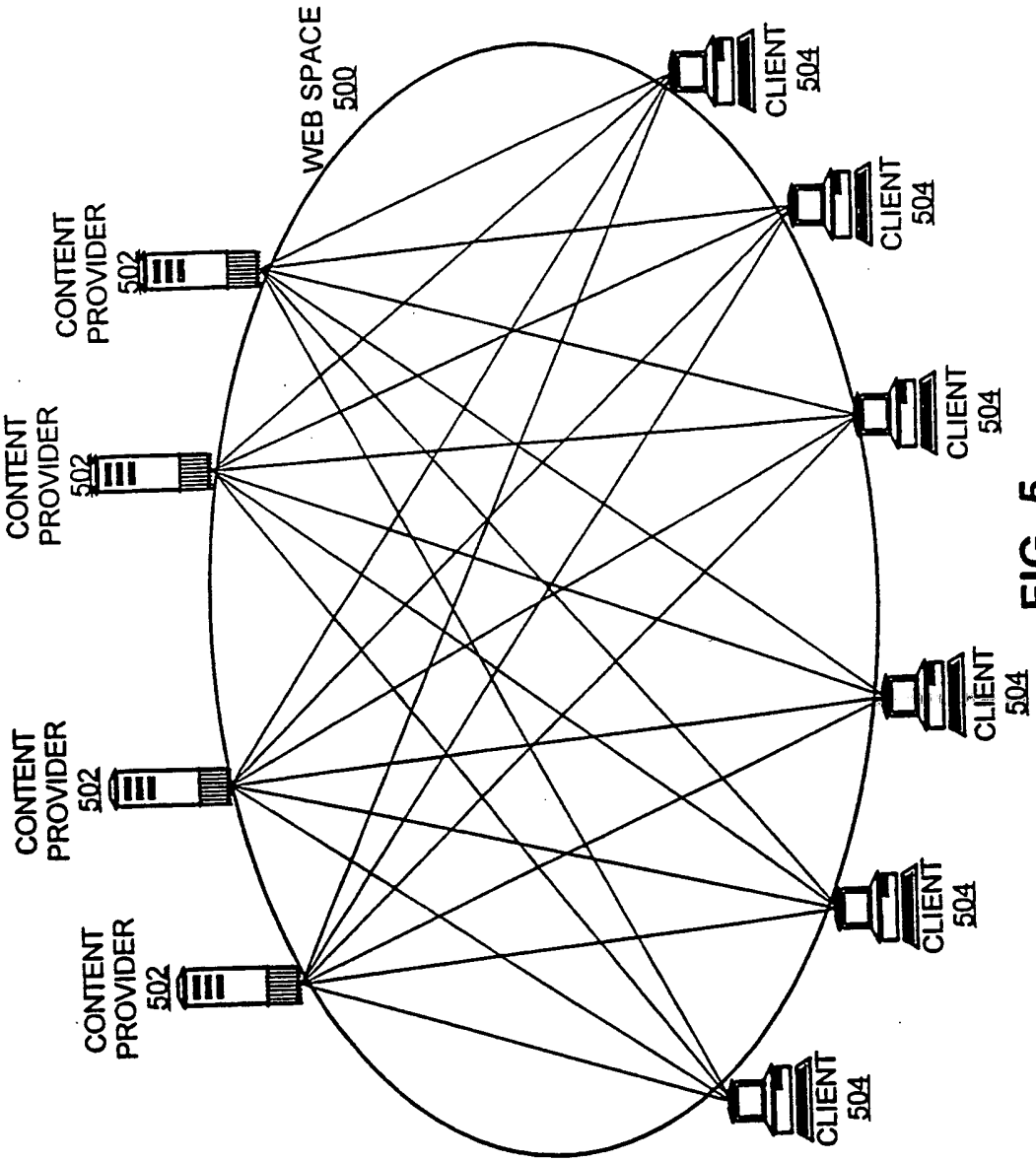


FIG. 5
(PRIOR ART)

INTERNATIONAL SEARCH REPORT

Interna[illegible]al Application No
PCT/US 00/22413

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/18 H04L29/06 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	RODRIGUEZ P ET AL: "Improving the WWW: caching or multicast?" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 30, no. 22-23, 25 November 1998 (1998-11-25), pages 2223-2243, XP004152174 ISSN: 0169-7552 page 2241, left-hand column, line 8 -right-hand column, line 25 figure 4	1,2,7,8, 12,13
Y	--- -/--	3-5,9, 10,14-17

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 November 2000	08/12/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl. Fax: (+31-70) 340-3016	Ströbeck, A.

2

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	TOUCH J ET AL: "LSAM proxy cache: a multicast distributed virtual cache" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 30, no. 22-23, 25 November 1998 (1998-11-25), pages 2245-2252, XP004152175 ISSN: 0169-7552 page 2246, right-hand column, line 7 -page 2247, right-hand column, line 23 figures 1,3,4 ---	1,6,7, 11-13
Y	US 5 553 083 A (MILLER C KENNETH) 3 September 1996 (1996-09-03) column 2, line 30 -column 3, line 3 ---	3-5,9, 10,14-17
A	KANCHANASUT K ET AL: "The AI3 CacheBone Project" PROCEEDINGS OF 1999 INTERNET WORKSHOP (WS'99), OSAKA, JAPAN, 18 - 25 February 1999, pages 203-208, XP002153935 Piscataway, NJ, USA page 204, right-hand column, line 11 -page 205, left-hand column, line 27 -----	4,10

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern ıal Application No
PCT/US 00/22413

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5553083 A	03-09-1996	AU 5295096 A	07-08-1996
		EP 0804838 A	05-11-1997
		JP 10512726 T	02-12-1998
		WO 9622641 A	25-07-1996
		US 5727002 A	10-03-1998
		US 5920701 A	06-07-1999

This Page Blank (uspto)